

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual professional practice in the company

Zadání bakalářské práce

Student: **Ivana Wzientková**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**
Individual Professional Practice in the Company

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: KVADOS, a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **doc. RNDr. Petr Šaloun, Ph.D.**


Konzultant bakalářské práce: Bc. Radek Pindora

Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry



prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně. Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

V Ostravě 18. Prosince 2017


.....

Souhlasím se zveřejněním této bakalářské/diplomové práce dle požadavků čl. 26 odst. 9.
Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-
TU Ostrava

V Ostravě 18. Prosince 2017


.....

Ráda bych na tomto místě poděkovala celému týmu programátorů, kteří mi velice pomáhali po celou dobu praxe a naučili mě vše potřebné, abych mohla samostatně pracovat na projektu. Speciální poděkování chci věnovat Josefu Šustkovi, který mi zodpovědně zadával práci a při každém problému mi nesmírně pomohl.

Abstrakt

Tato práce se zabývá individuální praxí ve společnosti KVADOS, a. s. Praxi jsem vykonávala od 1. září 2017 do 30. dubna 2018 na pozici testera, který se zabývá programováním testů uživatelského rozhraní. Na začátku popisuji společnost KVADOS a své působení v této společnosti. Zabývám se jednotlivými úkoly, které mi byly během praxe zadány a podrobně je popisuji. Dále popisuji dovednosti získané během praxe, a také dovednosti, které mi k vykonání praxe chyběly. V závěru zhodnocuji celé své působení ve firmě a dosažené výsledky.

Klíčová slova: UI testy, KVADOS a. s., myAVIS NG, Espresso, QAS, VENTUS

Abstract

This work is focused on individual practical experience in a company KVADOS a.s. The work has been carried out since 1st September 2017, till 30th April 2018 on a position of a tester, which deals with programing of user interface tests. At the beginning I'm describing the company of KVADOS, as well as my job in this company. Next, I'm dealing with individual tasks, which have been entrusted to me during my work while also specificaly describing them. Then, I describe abilities gained throughout my work and also abilities, which I lacked for this job. In conclusion I evaluate my whole performance in this company and my achieved outcomes.

Key Words: UI tests, KVADOS a. s., myAVIS NG, Espresso, QAS, VENTUS

Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	10
Seznam tabulek	11
Seznam výpisů zdrojového kódu	12
1 Úvod	13
2 Popis praxe ve společnosti KVADOS, a.s.	14
2.1 O společnosti KVADOS	14
2.2 Popis pracovního zaměření	14
2.3 O aplikaci myAVIS NG	14
3 Seznam zadaných úkolů v průběhu praxe	16
4 Technologie a postupy použité při bakalářské praxi	17
4.1 Espresso	17
4.2 QAS	17
4.3 VENTUS	18
5 Řešení zadaných úkolů	19
5.1 Základní příkazy pro tvorbu automatických testů	19
5.2 Školení ve firmě KVADOS a. s.	19
5.3 Scénář: Obchodní případy - kontrola hodnot na dlaždici a ve fialové liště	19
5.4 UI test: Obchodní případy - kontrola hodnot na dlaždici a ve fialové liště	21
5.5 UI test: Práce s průvodci - kontrola nedostupnosti průvodce	22
5.6 Plán návštěv - změna pořadí návštěv, kontrola správnosti časů	23
5.7 Katalog - Změna měrné jednotky sortimentu	25
5.8 Distribuce - kontrola ikon A,N,T	26
5.9 Pravidelná kontrola spouštění automatických testů	28
5.10 Interní postup k vyřešení chyb	28
6 Zhodnocení dosažených výsledků v průběhu odborné praxe	30
6.1 Získané znalosti a dovednosti v průběhu studia použité při bakalářské praxi	30
6.2 Scházející dovednosti v průběhu odborné praxe	30
7 Závěr	31

Seznam použitých zkratek a symbolů

UI	– User Interface
TFS	– Team Foundation Server
SQL	– Structured Query Language
API	– Application Programming Interface
JVM	– Java Virtual Machine
ERP	– Enterprise Resource Planning
GPS	– Global Positioning System

Seznam obrázků

1	Ventus	18
2	Dlaždice obchodních případů s měnícími se hodnotami	20
3	Fialová lišta uvnitř agendy obchodních případů	20
4	Testovací scénář obchodních případů	20
5	Agenda obchodní případy	21
6	Agenda plán návštěv	23
7	Agenda plán návštěv po přesunu prvku	24
8	Round picker v agendě katalogu	25
9	Zobrazení ikon a filtrů v distribuci	27
10	Log automatických testů po nočním spuštění testů	28
11	Seznam chyb v interním systému	29

Seznam tabulek

1	Názvy úkolů a časová náročnost jednotlivých úkolů	16
---	---	----

Seznam výpisů zdrojového kódu

1	Metoda pro návratovou hodnotu otevřeno z databáze	21
2	Automatický test na kontrolu nedostupnosti průvodce	22
3	Metoda pro uchycení posuvníku a přesun	23
4	Ukázka použití reflexe při nastavení měrné jednotky	26
5	Metoda pro kontrolu ikon v listu v distribuci	27

1 Úvod

Získávání praktických znalostí a začlenění se do firemního systému považuji za důležitou část vzdělávání se a přípravy na budoucí práci. Toto byl důvod, proč jsem se rozhodla zpracovat bakalářskou práci formou praxe ve firmě KVADOS a. s., kde jsem po přijímacím řízení byla přidělena do týmu programátorů pro android aplikace na pozici testera, který se zabývá psaním automatických testů. Tyto automatické testy šetří čas testerům, kteří procházejí aplikaci manuálně. Dále se také brzy odhalí chyby, které vznikají neustálým vývojem aplikace. Tímto se předchází chybám u zákazníka a snižují se i kapacity na zaměstnance na pozicích konzultantů.

2 Popis praxe ve společnosti KVADOS, a.s.

2.1 O společnosti KVADOS

[1] Společnost KVADOS a. s. se zabývá vývojem vlastních softwarových řešení, služeb a poradenstvím. Tato firma je na trhu již od roku 1992 a aktuálně působí ve dvanácti zemích světa. Na výrobě softwaru pracuje okolo 150 zaměstnanců. KVADOS a. s. pravidelně prochází certifikacemi podle mezinárodních norem ISO využívaných v odvětví informačních a komunikačních technologií.

Jako přední vývojář informačních systémů v Moravskoslezském kraji patří k základajícím členům sdružení IT Cluster. Dále je také členem Czech ICT Alliance a Centra pro rozvoj dopravních systémů RODOS. Mezi partnery společnosti patří například T-Mobile, ATComputers, Microsoft a spousta dalších.

KVADOS od roku 2010 vydává vlastní magazín yourK, který přináší zákazníkům a obchodním partnerům informace o společnosti, nových technologiích, a zejména témata a náměty, jak mohou zlepšit svůj byznys a procesy.

2.2 Popis pracovního zaměření

Od začátku praxe jsem pracovala v týmu pro android, který se zabýval vývojem aplikace myAVIS NG [2.3] pro mobilní zařízení. Hlavním úkolem bylo vytváření automatických testů, které testují rozhraní aplikace myAVIS NG, a zároveň i kontrolují testovací data, která jsou v aplikaci viditelná nebo se nachází v databázi. Testovací data se předem vytváří v aplikaci a poté se při synchronizaci odesílají na server nebo vytváří rovnou na serverové straně v ERP systému Ventus [4.3].

K této pozici patří i psaní scénářů pro automatické testy. Scénáře se píšou na základě analýzy k aplikaci myAVIS NG. Při psaní automatických testů a procházení aplikace se objevují občas chyby. Nalezené chyby je třeba zapsat do seznamu chyb [11] a předat je takto programátorovi.

2.3 O aplikaci myAVIS NG

[2] Aplikace myAVIS NG je aplikace pro obchodní zástupce, která jim pomáhá provádět návštěvy u zákazníka. Mezi nejzákladnější agendy patří například agenda zákazníků, ve které je seznam všech stávajících zákazníků, které obchodní zástupce navštěvuje.

Nad každým zákazníkem lze provádět spoustu akcí jako například zobrazit seznam pohledávek, smluv, zapůjčeného zařízení, věrnostních programů zákazníka, úkolů apod. Agenda rovněž obsahuje základní informace jako kontaktní údaje, provozní dobu nebo jiné kontaktní osoby.

Další důležitou agendou je plán návštěv, do kterého si může obchodní zástupce naplánovat, kdy navštíví své zákazníky, popřípadě si může návštěvy vygenerovat ze šablony, kterou si dříve vytvořil. V aplikaci je také agenda personalistika, do které si může obchodní zástupce vykazovat práci, popřípadě vkládat záznamy do knihy jízd.

Toto je jen náhled do aplikace myAVIS NG, jelikož aplikace má spoustu funkcí, které mají obchodnímu zástupci pomáhat, jako třeba katalog zboží, průvodce objednáváním a prodejem, seznam prodejních akcí apod. Každý zákazník společnosti KVADOS si může vybrat, které agendy bude využívat.

3 Seznam zadaných úkolů v průběhu praxe

V této sekci popisují jednotlivé úkoly a jejich časovou náročnost, která se zakládá na firemní kalkulaci pro jednotlivé úkoly. Kalkulace je orientační a úkol by měl být v této době splněn.

Název	Časová náročnost (h)
Školení ve firmě KVADOS a. s.	40
UI test: Distribuce - kontrola ikon A,N,T	8
Scénář: Obchodní případy - kontrola hodnot na dlaždici a ve fialové liště	8
UI test: Obchodní případy - kontrola hodnot na dlaždici a ve fialové liště	10
UI test: Katalog - kontrola ikon A,N,T	8
UI test: Katalog - změna měrné jednotky sortimentu	12
UI test: Plán návštěv - změna pořadí návštěv, kontrola správnosti časů	10
UI test: Obchodní případy - kontrola stavu obchodního případu	8
UI test: Výkaz práce	6
UI test: Kniha jízd	6
UI test: Práce s průvodci - kontrola nedostupnosti průvodce	6
Pravidelná kontrola spouštění automatických testů	neomezeno
UI test: Dlaždicové zobrazení katalogu	10
UI test: Druhotné umístění v průvodci	14
UI test: Editace zařízení	10

Tabulka 1: Názvy úkolů a časová náročnost jednotlivých úkolů

4 Technologie a postupy použité při bakalářské praxi

4.1 Espresso

[3] Espresso je knihovna pro podporu testování pro Android, přes rozhraní API na psaní automatických testů pro uživatelské rozhraní. Tato knihovna se používá na zařízeních se systémem Android 2.3.3 a vyšší. Hlavní výhodou používání této knihovny spočívá v tom, že poskytuje automatickou synchronizaci testovacích akcí s uživatelským rozhraním aplikace, kterou testujete.

Tato knihovna zjistí, kdy je hlavní vlákno nečinné, takže je schopná spustit příkazy v testech ve vhodném čase, a zajistit tak vyšší spolehlivost testů. Espresso je rozhraní API založené na přístrojovém vybavení, a pracuje s testovacím systémem AndroidJUnitRunner.

AndroidJUnitRunner je nástroj pro běh automatických testů. To je v podstatě vstupní bod pro provoz celého souboru testů. Ovládá testovací prostředí a spouští všechny testy definované v testovacím balíčku. Ve vhodném souboru AndroidManifest.xml je definován přístroj s androidem: název nastavený na AndroidJUnitRunner a příslušnou sadu android: targetPackage.

Možnosti provedení:

- Spuštění všech testů:
`adb shell am instrument -w cesta_k_balíčku_testů/cesta_k_AndroidJUnitRunner`
- Spuštění všech testů ve třídě: `adb shell am instrument -w -e class cesta_ke_třídě cesta_k_balíčku_testů/cesta_k_AndroidJUnitRunner`
- Spuštění jediného testu: `adb shell am instrument -w -e class cesta_ke_třídě#jmeno_testu cesta_k_balíčku_testů/cesta_k_AndroidJUnitRunner`

4.2 QAS

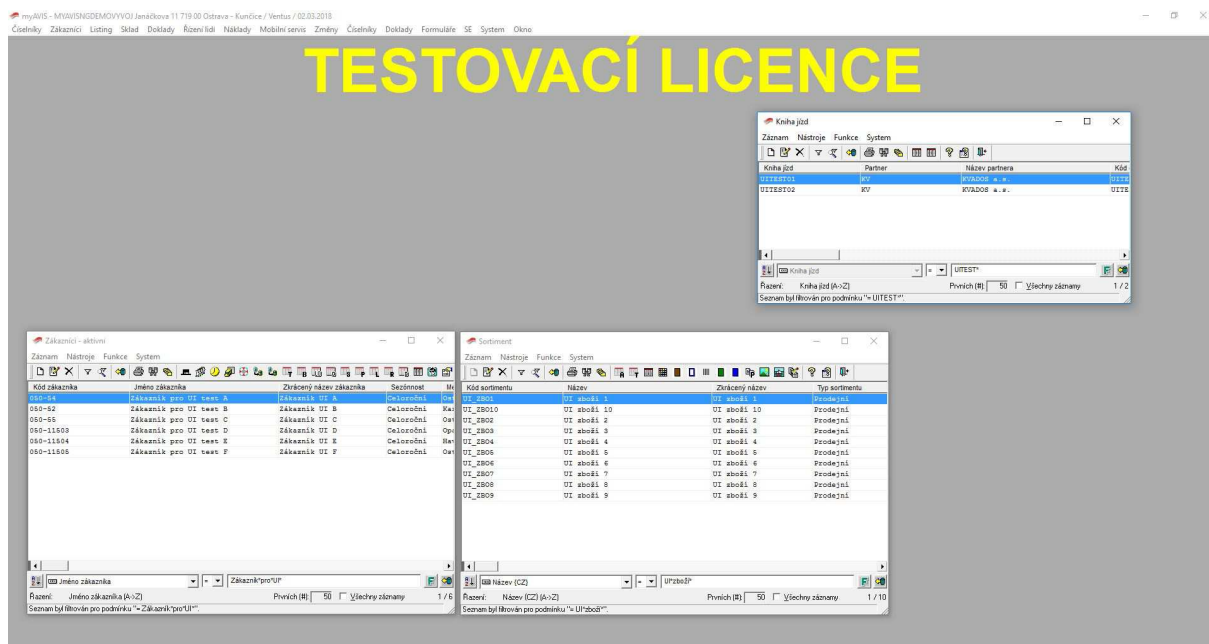
Jádro aplikace myAVIS NG. Do projektu je importováno jako knihovna. Nacházejí se zde základní komponenty, které se používají pro zobrazení dat. Dále jsou zde metody a funkce pro synchronizaci dat a pro práci s databází. Toto jádro obsahuje také aktivity z androidu. Mimo jiné obsahuje také metody pro práci s fotoaparátem, úložištěm dat v zařízení, a práce s GPS souřadnicemi.

Také se zde nachází základní třída pro tvorbu automatických testů, která obsahuje metody na počáteční přihlášení do aplikace před spuštěním samotného testu. Veškeré mnou psané testy dědí právě z této třídy.

4.3 VENTUS

[4] Interní ERP systém společnosti KVADOS a. s. Tento systém je rozdělený na moduly, které jsou přiřazeny k jednotlivým projektům. V těchto modulech je možné nastavovat data pro aplikaci, editovat tyto data a přiřazovat je k jednotlivým divizím a teritoriím.

Ve chvíli, kdy pro svůj test potřebuji vytvořit data, přihlásím se do systému VENTUS. Zde tato data vytvořím a přiřadím si je ke svému zařízení. Při spuštění synchronizace na zařízení si aplikace myAVIS NG stáhne data ze serveru VENTUS a uloží je do interní SQLite databáze.



Obrázek 1: Ventus

5 Řešení zadaných úkolů

5.1 Základní příkazy pro tvorbu automatických testů

Pro tvorbu automatických testů se používá pár základních příkazů, které zajistí, aby aplikace simulovala klikání na prvky, jako by to dělal člověk sám.

Základní příkazy:

- `onView(withUIComponent("nazev_komponenty")).perform(scrollTo(), click());` - tento příkaz umožní posunutí ke komponentě a poté na ní klikne podle jejího názvu
- `onData(isNth(cislo_prvku)).inAdapterView(withUIComponent"nazev_listu")).perform(click());` - tento příkaz zajistí kliknutí na prvek v listu, přičemž v metodě `isNth()` určíme na který prvek má kliknout podle pořadí
- `onView(isRoot()).perform(swipeLeft());` - simuluje posun prstu po obrazovce zprava doleva
- `pressBack();` - klikne na tlačítko zpět

5.2 Školení ve firmě KVADOS a. s.

Ihned po nástupu jsem dostala firemní notebook a veškeré potřebné vybavení k práci. Po nainstalování všech programů jsem byla seznámená s prostředím Android studia, Visual Studio a s projektem myAVIS NG. Byly mi ukázány scénáře, a poté základní psaní automatických testů. Následně jsem začala psát své první testy.

Ve firmě se používá interní personalistika pro výkaz práce, do které jsem také byla zaškolená. Následně proběhlo školení o bezpečnosti práce na pracovišti. Během následujících týdnů jsem byla podrobně seznámená s aplikací, jak přesně funguje, a také jsem byla zaškolená na interní systém VENTUS [4.3](#) a vytváření dat v něm.

5.3 Scénář: Obchodní případy - kontrola hodnot na dlaždici a ve fialové liště

Scénáře pro UI testy obvykle píše zkušenější tester, ale pokud je zaneprázdněn důležitější prací, je třeba, abych si scénář napsala sama. Tento scénář byl kritický pro klienty a bylo velice důležité kontrolovat po každé změně v datech změnu na dlaždici na obrázku [2](#) a v informační (fialové) liště na obrázku [3](#).

Každý obchodní případ je reprezentován různými stavy. Mezi tyto stavy patří:

- Otevřeno - reprezentuje otevřený obchodní případ.
- Po termínu - obchodní případ, který nebyl uzavřen před koncem termínu.
- Získáno - obchodní případ, který skončil pozitivně.

- Ztraceno - reprezentuje případ, který nebyl z nějakého důvodu dokončen.

Tyto stavy se postupně zobrazují na dlaždici v hlavním menu. Pokud rozklikneme agendu obchodních případů, zobrazí se informativní fialová lišta, ve které se kromě popsanych stavů zobrazuje navíc ještě stav uzavřeno, který reprezentuje počet uzavřených případů a hodnota celkem, ve které se zobrazuje celkový počet všech obchodních případů. Každý obchodní případ je také přiřazen ke svému zákazníkovi. I toto bylo velmi důležité kontrolovat.

Obchodní případy	Obchodní případy	Obchodní případy	Obchodní případy
1 získáno	2 ztraceno	10 otevřeno	5 po termínu

Obrázek 2: Dlaždice obchodních případů s měnícími se hodnotami

Celkem 14 • Otevřeno 10 • Uzavřeno 4 • Získáno 1 • Ztraceno 2

Obrázek 3: Fialová lišta uvnitř agendy obchodních případů



Obrázek 4: Testovací scénář obchodních případů

5.4 UI test: Obchodní případy - kontrola hodnot na dlaždici a ve fialové liště

Tento automatický test byl vytvořen na základě scénáře, který byl popsán výše v kapitole 5.3. V tomto testu se musí kontrolovat jak hodnoty, které jsou viditelné v aplikaci na dlaždici, tak se musí kontrolovat i hodnoty zapsané aktuálně v databázi. Jako příklad metody pro přístup do databáze uvádím metodu pro návratovou hodnotu otevřeno 1.

V testu se vytváří několik nových obchodních případů, a poté se mění jejich stav. Po každé změně se kontroluje jak fialová lišta, tak i dlaždice a hodnoty v databázi, kdy se všechny hodnoty musí rovnat. V agendě je každý obchodní případ prezentován jedním prvkem v listu a počet prvků v listu se rovná poli "Celkem" na fialové liště. Celá agenda je vidět na obrázku číslo 5.

myAVIS® NG Obchodní případy

Celkem 14 • Otevřeno 10 • Uzavřeno 4 • Získáno 1 • Ztraceno 2					
Vše	Stav	Typ	Fáze	Název	
UI případ	UI zákazník jiný • Dana Jur...	Hejtmanova 22/9 Karviná	POP materiál UI	UI fáze server do 31. 12. 2078	
UI případ	Zákazník pro UI test D •	Opavská 3/12 Opava	POS materiál UI test		
UI příležitost	Zákazník pro UI test B • Ro...	Karvinská 110 Karviná	UI nenaskladnit	UI fáze konec	
UI případ	Zákazník pro UI test A • Zd...	Novoveská 1139/22 Ostrava	UI test	UI fáze začátek do 27. 07. 2016	!

Obrázek 5: Agenda obchodní případy

```
private String getOpenedValueString()
{
    AdvanceQuery advanceQuery;
    IntegerEntity integerEntity;
    advanceQuery = MblBusinessCasesHelper.getBusinessCasesInfoOpenCountQuery
        (); //opened
    integerEntity = SqlHelper.getEntityFromQuery(advanceQuery, IntegerEntity.
        class);
    if (integerEntity != null && integerEntity.getValue() != null)
```

```

{
    return String.valueOf(integerEntity.getValue());
}
return "0";
}

```

Výpis 1: Metoda pro návratovou hodnotu otevřeno z databáze

5.5 UI test: Práce s průvodci - kontrola nedostupnosti průvodce

Průvodce je posloupnost jednotlivých kroků, které napomáhají obchodním zástupcům jednotlivých firem získat veškeré informace během návštěvy u svých zákazníků.

Na serveru se mohou definovat průvodci, kteří jsou přiřazeni k jednotlivým divizím a rolím. Divize označuje oblast, do které jednotliví pracovníci spadají. Takto je můžeme rozdělit například do regionálních oblastí, ve kterých pracují nebo pro různé typy zákazníků, pro které pracují např. Slovensko a Česká republika. Role označuje pozici, na které je pracovník např. obchodní zástupce, ředitel apod.

Toto rozdělení zajišťuje, že každému pracovníkovi se zobrazují jen ti průvodci, které potřebuje ke své práci. Také se může na serveru definovat průvodce, který není viditelný ještě nikomu. Jednoduchý automatický test [2](#) kontroluje na třech příkladech průvodců, zdali se nezobrazují na testovacím zařízení.

```

@Test
public void unpublished()
{
    onView(withUIComponent("Customers")).perform(scrollTo(), click());
    onData(isFirst()).inAdapterView(withUIComponent("List")).perform(
        longClick());

    assertNull(getActionflowSpecificationEntity("UI_ROLE_COND"));
    assertNull(getActionflowSpecificationEntity("UI_UNPUBLISHED"));
    assertNull(getActionflowSpecificationEntity("UI_DIVISION_COND"));
}

```

Výpis 2: Automatický test na kontrolu nedostupnosti průvodce

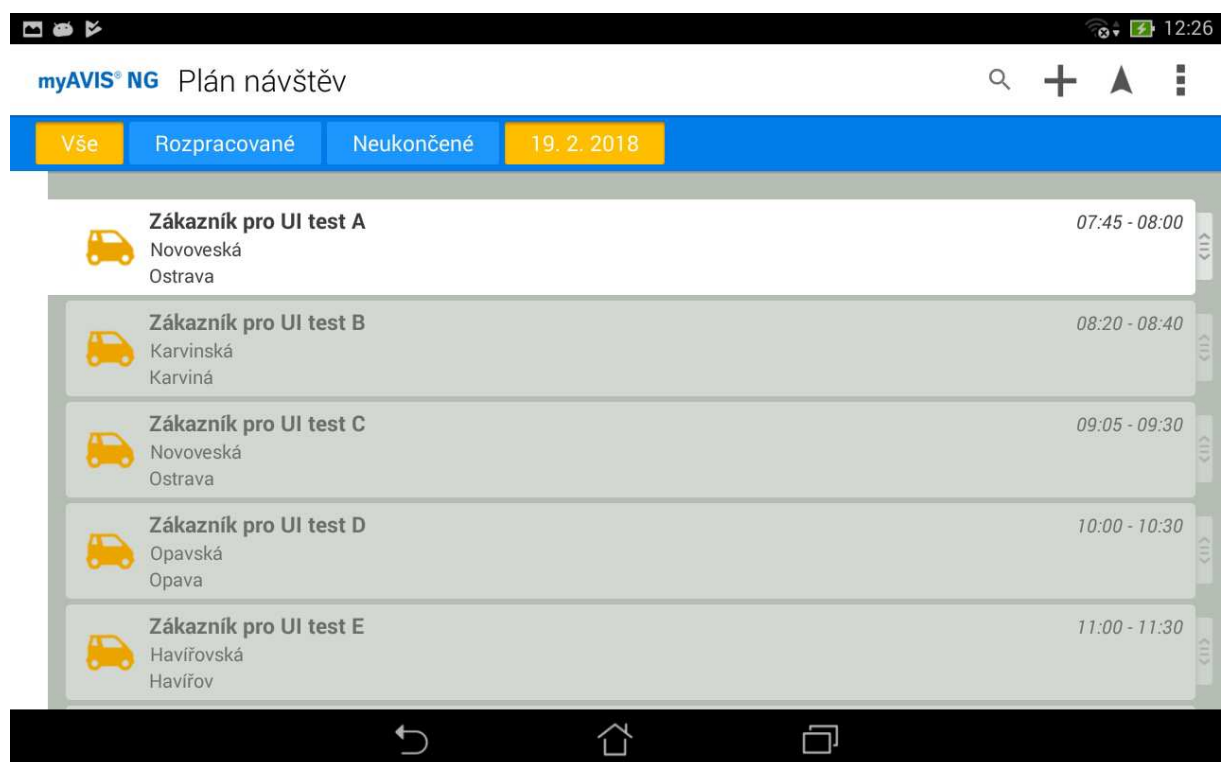
Na začátku testu se otevře seznam všech zákazníků. Poté se vybere první zákazník ze seznamu, dlouhým přidržením na zákazníkovi se zobrazí tabulka dostupných průvodců. Následně

se na základě názvu průvodce v databázi kontroluje, zda-li není viditelný. V případě, že by viditelný byl, test by neprošel.

5.6 Plán návštěv - změna pořadí návštěv, kontrola správnosti časů

Plán návštěv je agenda, která obsahuje seznam zákazníků, které má obchodní zástupce v daný den navštívit. Na začátku testu je třeba vygenerovat seznam návštěv ze šablony, která už je předem nadefinovaná.

Každý řádek seznamu má svůj posuvník, kterým lze měnit pořadí návštěv. Tyto posuvníky lze vidět na obrázku [6](#). Tento posun se provádí v testu pomocí metody [3](#). Tato metoda zajišťuje přesun prvku z pozice 5 na pozici 2.



Obrázek 6: Agenda plán návštěv

```
int location[] = new int[2];  
View view1 = findViewById(isNthMatching(withId(R.id.drag_handle), 5));  
view1.getLocationOnScreen(location);  
float fromX = (float) location[0];  
fromX += 5;  
float fromY = (float) location[1];
```

```

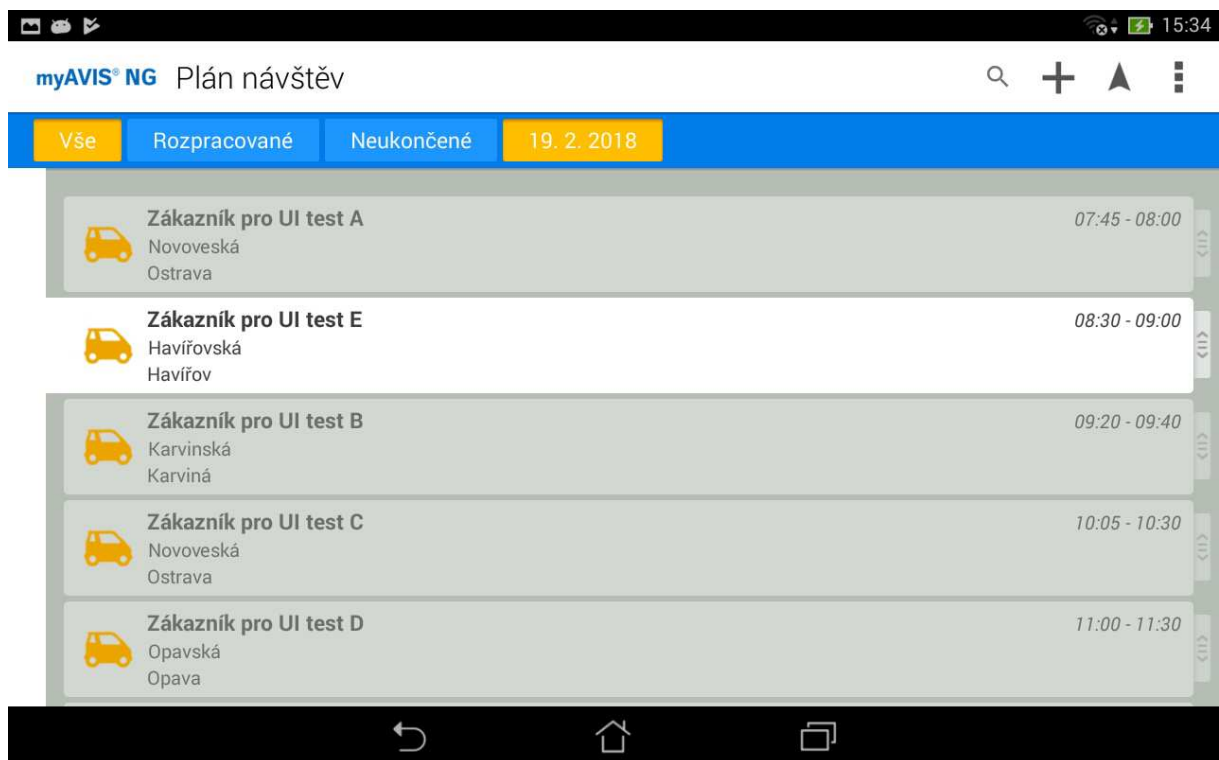
fromY += 10;
view1 = findViewById(isNthMatching(withId(R.id.drag_handle), 2));
view1.getLocationOnScreen(location);
float toX = (float) location[0];
toX += 5;
float toY = (float) location[1];
toY += 10;

TouchUtils.drag(getTestRunner(), fromX, toX, fromY, toY, 20);

```

Výpis 3: Metoda pro uchycení posuvníku a přesun

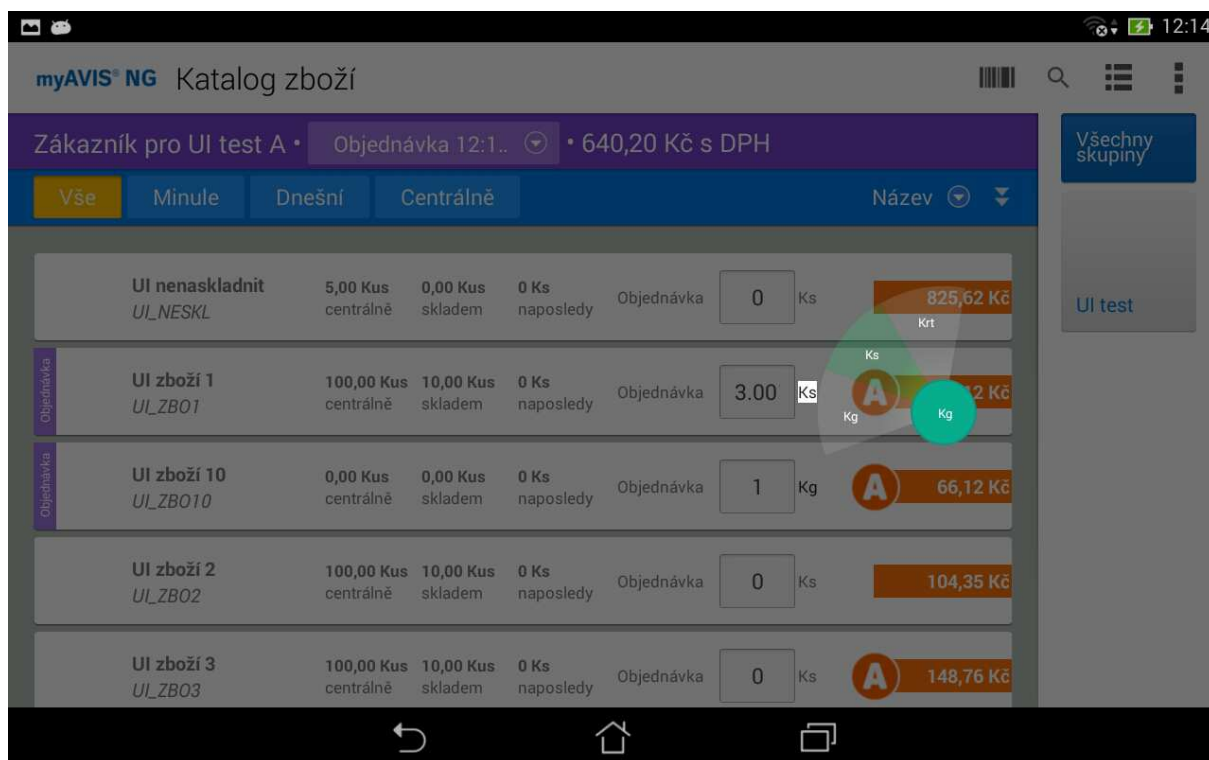
Každá návštěva obsahuje čas naplánované návštěvy, který značí od kdy, do kdy by měla návštěva u zákazníka probíhat. Při změně pořadí návštěv se přepočítává i čas, tak jak je to vidět na obrázku 7. Tyto časy se potom musí kontrolovat v testu.



Obrázek 7: Agenda plán návštěv po přesunu prvku

5.7 Katalog - Změna měrné jednotky sortimentu

Katalog je agenda, ve které si můžete podobně jako v jakémkoliv prodejním katalogu vybrat sortiment, který si zároveň můžete i rovnou objednat. Každý řádek sortimentu obsahuje své textové pole pro vyplnění množství. Zároveň některé sortimenty obsahují tabulku pro konverzi měrných jednotek, tím pádem můžeme změnit měrnou jednotku z kusů třeba na kilogramy. K tomuto výběru se používá prvek tzv. round picker [8](#).



Obrázek 8: Round picker v agendě katalogu

Tento round picker se zobrazí při dlouhém stisku na měrné jednotce a pomocí pohybu po obrazovce se vybere výše s měrnou jednotkou, kterou uživatel potřebuje. Toto gesto by v automatických testech mohlo dělat problém, proto jsem zvolila jako řešení nastavení přesné hodnoty.

Metoda, která tuto měrnou jednotku nastavuje, je v jádře QAS a navíc privátní. Tímto jsem se dostala k využití reflexe v Javě viz [4](#). Nejprve je třeba najít daný round picker a poté vytvořit nové vlákno, ve kterém se změní jednotka z kusů na kartony. Poté se vyhledá privátní metoda a povolí se k ní přístup jako by byla veřejná a poté se tato metoda zavolá.

```

final AssortmentRoundPicker assortmentRoundPicker = findViewById(R.id.
    myavis_sfa_assortment_detail_order_unit_picker));
name = assortmentRoundPicker.getText().toString();
assertEquals("Ks", name);
getCurrentActivity().runOnUiThread(new Runnable()
{
    @Override
    public void run()
    {
        assortmentRoundPicker.setValue("KAR");
        Method fireOnValueChanged = ReflectionUtils.findMethod(RoundPicker.
            class, "fireOnValueChanged");
        fireOnValueChanged.setAccessible(true);
        ReflectionUtils.invokeMethod(fireOnValueChanged,
            assortmentRoundPicker);
    }
});

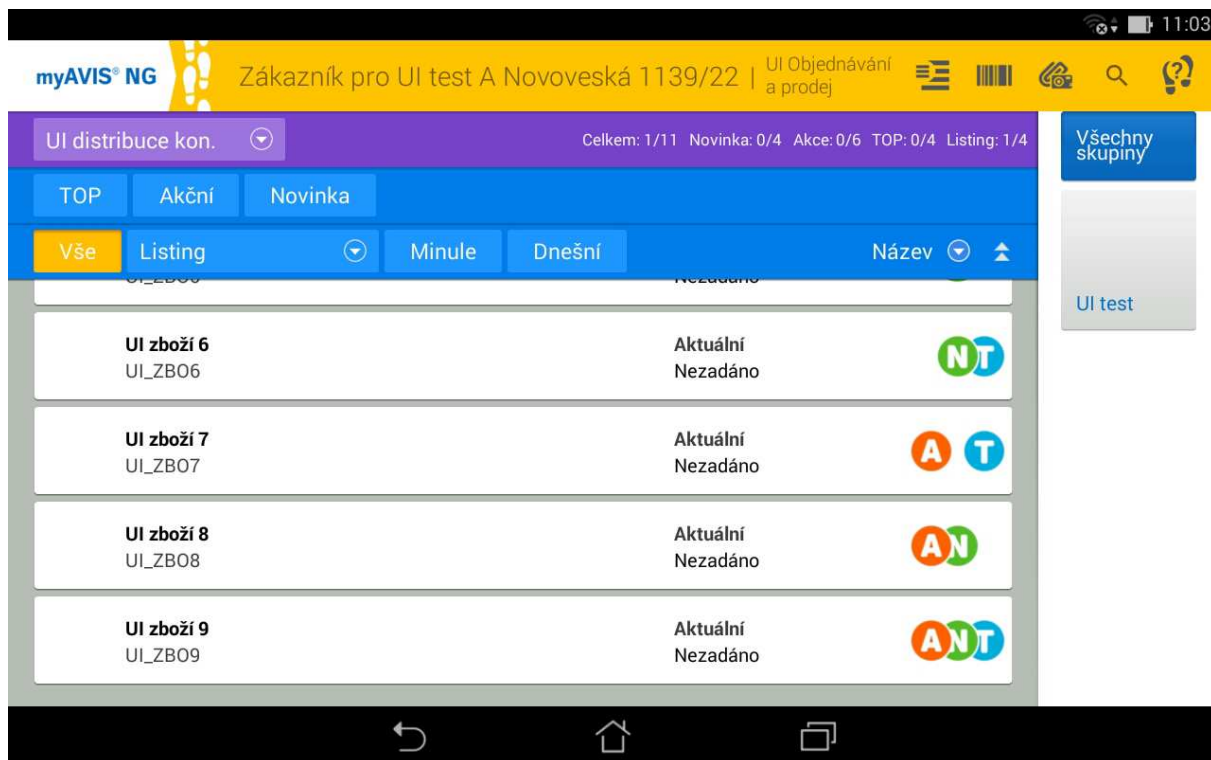
```

Výpis 4: Ukázka použití reflexe při nastavení měrné jednotky

Automatický test na změnu měrné jednotky změní několikrát měrnou jednotku a kontroluje, zda-li se dobře přepočítává zadané množství podle konverzní tabulky. Zároveň se v testu kontroluje propočítávání ceny za celkovou objednávku, která je viditelná ve fialové liště na obrázku [8](#).

5.8 Distribuce - kontrola ikon A,N,T

Každý sortiment v distribuci může obsahovat ikonu A (Akční), N (Novinka), T (TOP) nebo jejich kombinace. Dle těchto ikon lze použít také filtr, který nám může vyfiltrovat například novinky v sortimentu. Při testu se postupně kliká na filtry, které jsou zobrazeny na obrázku [9](#), a kontroluje se, zda-li zobrazené zboží obsahuje ikonu, která patří k těmto filtrům. Zároveň se musí zobrazovat i ostatní ikony, které sortiment zobrazuje. Pro tuto kontrolu jsem vytvořila metodu [5](#).



Obrázek 9: Zobrazení ikon a filtrů v distribuci

```
public void controlIconInDistr(boolean action, boolean nev, boolean top, int
    position)
{
    DecorateRecyclerView list = findViewById(withUIComponent("List"));
    DistributionAssortmentsRecyclerViewAdapter recyclerAdapter = (
        DistributionAssortmentsRecyclerViewAdapter) list.getAdapter();
    DistributionListDataObject object = (DistributionListDataObject)
        recyclerAdapter.getObjects().get(position);
    boolean iconAction = object.isAction() == View.VISIBLE;
    assertEquals(iconAction, action);
    boolean iconTop;
    if (object.getTopVisible() == 0)
    {
        iconTop = true;
    }
    else
    {
        iconTop = false;
    }
}
```

```

    }
    assertEquals(iconTop, top);
    boolean iconNew = object.isNew() == View.VISIBLE;
    assertEquals(iconNew, nev);
}

```

Výpis 5: Metoda pro kontrolu ikon v listu v distribuci

5.9 Pravidelná kontrola spouštění automatických testů

Automatické testy jsou ukládány na serveru, ze kterého se pravidelně v noci spouští. Na začátku každého pracovního dne přichází čas na kontrolu průběhu testů. Pokud se během posledního pracovního dne vytvořily nějaké chyby v aplikaci, testy je odhalí a okamžitě se zadávají k opravě programátorům.

Jako další chyby se mohou projevit změny v databázovém modelu nebo chyby v datech obecně. Na obrázku [10](#) je vidět log, ve kterém zelené testy označují testy, které prošly bez chyby a červené jsou testy s chybou.



Obrázek 10: Log automatických testů po nočním spuštění testů

Tento log jde dále rozkliknout a tím zobrazit, ve které třídě a na jakém řádku test neprošel. Ve chvíli pádu testu se udělá také poslední snímek obrazovky, pro lehčí orientaci v aplikaci. Toto urychlí vyhledávání problémů. Každý test, který nebyl dokončen úspěšně, je třeba zkontrolovat a zjistit příčinu chyby.

Kromě logu na obrázku [10](#) probíhají i další dva průběhy testů, ze kterých je také vytvořen log, dohromady je těchto testů kolem 200 a jsou rozděleny do 3 průběhů. Toto rozdělení bylo důležité, jelikož při běhu 150 testů naráz začínala pravidelně docházet paměť v zařízení.

5.10 Interní postup k vyřešení chyb

Při nalezení chyby se nejprve prochází analýzy, ve kterých se kontroluje, zda-li se jedná o chybu, či se změnila funkčnost. Pokud se potvrdí, že se jedná o chybu, tak se tato chyba zapíše do interního seznamu chyb [11](#).

➤ Přidat novou položku

<input type="checkbox"/>	ID problému	Nadpis	Přířazeno	Stav problému	Priorita	Požadovaný termín splnění	Číslo požadavku	Příčina chyby dle konzultanta	Autor	Oblast
	1481	Pád aplikace při kliku na výběr datumů ve filtrační liště v personalistice	Fichna Lukáš	Vyřešen v projektu	(1) Vysoká			Chyba v kódu	Wzientková Ivana	Standard
	1463	Pád aplikace v průvodci zásobováním	Fichna Lukáš	Vyřešen v projektu	(1) Vysoká			Chyba v kódu	Wzientková Ivana	Průzkumy
	1393	Pád aplikace v průvodci objednávání a prodej	Kovář Martin	Zadán	(1) Vysoká			Chyba v kódu	Wzientková Ivana	Objednávání a prodej
	1482	Není vidět poslední faktura v historii prodejů a objednávek	Grygar David	Vyřešen v projektu	(2) Normální			Chyba oproti zadání	Wzientková Ivana	Objednávání a prodej
	1478	Chybí informace na kontaktu v průvodci	Grygar David	Vyřešen v projektu	(2) Normální			Chyba oproti zadání	Wzientková Ivana	Zákazníci
	1154	Kontrola datových polí EEa- eea typu number v průvodci	Brecher Jakub	Vyřešen v projektu	(2) Normální			Neuvezeno	Wzientková Ivana	Standard
	1487	Chybějící refresh v pokladních dokladech	Grygar David	Zadán	(3) Nízká			Chyba oproti zadání	Wzientková Ivana	Pokladna
	1408	Filtrování podle "Datum dokladu" v pokladně	Fichna Lukáš	Vyřešen v projektu	(3) Nízká			Chyba oproti zadání	Wzientková Ivana	Pokladna

Obrázek 11: Seznam chyb v interním systému

Do tohoto seznamu se přidává zápis o chybě přes novou položku. Zde se vyplní nadpis chyby, popis chyby a priorita. Priorita 1 označuje závažnou chybu, jako je například pád celé aplikace. Priorita 3 označuje chybu, která není tak závažná a neovlivňuje tolik stěžejní funkce aplikace například špatné filtrování v aplikaci. Poté se chyba přiřadí na programátora, který bude chybu řešit. Také se zapíše oblast, ve které se chyba nachází, a o jaký druh chyby se jedná.

Druhy chyb:

- Nezjištěno - pokud nelze vybrat žádná z níže uvedených voleb
- Chyba konzultanta - chyby v datech, nepochopení - zapomenutí zadání, zásah do datového modelu.
- Chyba zákazníka - chyby v datech, nepochopení - zapomenutí zadání, zásah do datového modelu.
- Chyba v kódu - programátorská - výjimky, ostatní chyby v kódu.
- Chyba oproti zadání - kód funguje bez chyb, ale ne tak, jak by měl.
- Chyba nasazení - změna verze, aktivace nových funkcí, závislosti, konfigurace serveru, restartování služeb.
- Upřesnění zadání - nejde o chybu vzhledem k zadání, přesto je potřeba opravit.

6 Zhodnocení dosažených výsledků v průběhu odborné praxe

6.1 Získané znalosti a dovednosti v průběhu studia použité při bakalářské praxi

Základní znalost, kterou jsem použila při bakalářské praxi je znalost jazyka Java z předmětu Programovací jazyky 1. Dále jsem také využila SQL, při vytváření a kontrole dat na serveru, jehož znalost jsem získala z předmětu Úvod do databázových systémů, Databázové a informační systémy, a také z předmětu Vývoj informačních systémů.

Ke čtení analýz mi pomohly znalosti z předmětu Úvod do softwarového inženýrství. Ve 3. ročníku, jsem si za účelem seznámení se s vývojovým prostředím Android Studio a technologií Android, vybrala předmět Tvorba aplikací pro mobilní zařízení II.

6.2 Scházející dovednosti v průběhu odborné praxe

Při výkonu praxe mi scházely znalosti z předmětu Testování a softwarová kvalita, týkající se automatických testů, který je bohužel zahrnut až v navazujícím magisterském studiu. Dále taky znalosti o technologii Smart Client od Microsoftu, kterou jsem v průběh studia ve studijním plánu nezaznamenala, a se kterou jsem na praxi okrajově pracovala. V neposlední řadě mi chyběla znalost verzovacího systému TFS od Microsoftu. Všechny tyto znalosti jsem si v průběhu praxe samostudiem doplnila na úroveň, která byla potřebná ke kvalitnímu vykonání veškeré zadané práce.

7 Závěr

Závěrem chci říct, že bakalářskou praxi hodnotím velice kladně. Kolektiv je přátelský a atmosféra ve firmě se nese v rodinném duchu. Zaškolení na pozici probíhalo kvalitně a klidně. Během mé praxe jsem vytvořila přes 100 automatických testů, které pravidelně protestovávají aplikaci, a tímto předchází chybám u zákazníka. Zároveň jsem dostala i možnost předat své zkušenosti a znalosti novému kolegovi, který k nám nastoupil v únoru 2018, a který se zaškoluje na pozici programátora, a zároveň se bude starat o automatické testy.

Společnost KVADOS mi nabídla další spolupráci, a to na pozici testera pro manuální a automatické testy na plný úvazek. Jsem velice ráda, že jsem mohla vypracovat bakalářskou práci formou praxe, a tím si vyzkoušet reálnou práci ve firmě, absolvovat standardní pohovor a řešit reálné problémy, které mohou nastat během vykonávání práce.

Literatura

- [1] Společnost KVADOS [online]. [cit. 2018-01-04].

Dostupné z: www.kvados.cz

- [2] Produkt myAVIS NG [online]. [cit. 2018-01-04].

Dostupné z: www.kvados.cz/produkty/myavis-ng/

- [3] Knihovna espresso [online]. [cit. 2018-01-13].

Dostupné z:

<https://developer.android.com/training/testing/ui-testing/espresso-testing.html>

- [4] Produkt VENTUS [online]. [cit. 2018-01-20].

Dostupné z: www.kvados.cz/produkty/ventus/